

What is claimed is:

1 1. A method of performing a join in a database system comprising:
2 receiving a join query containing at least one function selected from the
3 group consisting of a selection predicate applied on a complex attribute, a projection
4 applied on a complex attribute, and a user-defined data type method;
5 determining a cost associated with applying the function on a first table
6 and a cost associated with applying the function on a second table; and
7 selecting a join path based on relative costs of applying the function on the
8 first and second tables.

1 2. The method of claim 1, wherein selecting the join path comprises applying
2 the function on one of the first and second tables associated with a lower cost.

3 3. The method of claim 1, wherein determining the costs comprises
4 determining the respective cardinalities of the first and second tables.

5 4. The method of claim 3, wherein determining the cost of applying the
6 function on the second table comprises determining the cost of a join table that is a result
7 of a join of the first table and another table.

1 5. The method of claim 3, wherein selecting the join path comprises applying
2 the function on one of the first and second tables that has the lower cardinality.

1 6. The method of claim 5, wherein the function comprises a selection
2 predicate applied on a complex attribute of the first table, the join query further
3 containing a projection applied on a complex attribute of the first table, the method
4 further comprising:
5 determining a cost associated with applying the projection on the first
6 table and a cost associated with applying the projection on the join table,
7 wherein selecting the join path comprises applying the projection on one
8 of the first table and the join table associated with a lower cost.

1 7. The method of claim 6, wherein selecting the join path comprises applying
2 the projection on one of the first table and join table with the lower cardinality.

1 8. The method of claim 1, further comprising identifying the function as a
2 costly function.

1 9. The method of claim 1, wherein the receiving, determining, and selecting
2 acts are performed by an optimizer module.

1 10. The method of claim 1, wherein determining the costs of applying the
2 function on the first and second tables comprises determining the costs of applying the
3 function on object relational tables.

1 11. An article comprising at least one storage medium containing instructions
2 that when executed cause a database system to:
3 receive a join query containing at least one function selected from the
4 group consisting of a selection predicate applied on a complex attribute, a projection
5 applied on a complex attribute, and a user-defined data type method; and
6 determine a join path for the join query based at least in part on a cost
7 associated with application of the function on the complex attribute.

12. The article of claim 11, wherein the join query specifies the function being applied on a first table, and wherein the instructions when executed cause the database system to determine the join path by applying the function on a second table different from the first table.

13. The article of claim 12, wherein the second table is a result of a join of the first table and another table.

14. The article of claim 11, wherein the join query specifies the function being applied on a first table, and
wherein the instructions when executed cause the database system to determine the join path by applying the function on a second table having a lower cardinality than the first table.

15. The article of claim 11, wherein the instructions when executed cause the system to determine the join path by applying the function on one of a first table and second table having a lower cardinality.

16. The article of claim 15, wherein the second table is a join of the first table and another table, and wherein the instructions when executed cause the system to determine the join path by further specifying a join of the second table and a third table to produce a fourth table.

17. The article of claim 16, wherein the join query further specifies application of a second function selected from the group consisting a selection predicate applied on a complex attribute, a projection applied on a complex attribute, and a user-defined data type method, the second function being applied on a third table,
wherein the instructions when executed cause the database system to determine the join path by further applying the second function on one of the third table and a fourth table with a lower cardinality,
wherein the fourth table is a join result of the third table and another table.

1 18. A database system comprising:
2 a storage system to store tables; and
3 an optimizer to receive a join query that specifies a function selected from
4 the group consisting of a selection predicate applied on a complex attribute, a projection
5 applied on a complex attribute, and a user-defined data type method,
6 the optimizer adapted to select a join plan based at least in part on a
7 comparison of a first cost of applying the function on a first table and a second cost of
8 applying the function on a second table.

1 19. The database system of claim 18, wherein the optimizer is adapted to
2 select the join plan that applies the function on the one of the first table and second table
3 with a lower cardinality.

4 20. The database system of claim 19, wherein the second table is a join result
5 of the first table and another table.

6 21. The database system of claim 20, wherein the join query specifies the
7 function being applied on the first table.

8 22. The database system of claim 20, wherein the first and second tables are
9 object relational tables.

1 23. The database system of claim 19, wherein the join query further specifies
2 application of a second function selected from the group consisting of a selection
3 predicate applied on a complex attribute, a projection applied on a complex attribute, and
4 a user-defined data type method, the join query specifying the second function being
5 applied on a third table, the optimizer adapted to select a join plan that applies the second
6 function on one of the third table and a fourth table with a lower cardinality, the fourth
7 table being a join result of the third table and another table.

1 24. The database system of claim 18, wherein the tables comprise object
2 relational tables.

1005395001